

REMARKS

The abstract has been amended. An abstract on a separate sheet is enclosed.

Claims 1-9 stand rejected under 35 USC 101 as being directed to non-statutory subject matter. Claims 1-9 have been amended so that they are now directed to a computer readable medium. Accordingly, the rejection has been overcome.

Claim 5 has been amended to overcome the rejection under 35 USC 112, second paragraph.

The independent claims 1, 10, 11, 12 and 21 stand rejected under 35 USC 102 over Altberg et al. Claim 1, as now amended, recites that the executable application recorded on the computer readable medium includes three components. These components are a program, one or more sub-routines in encrypted form (defined in the specification as including compression) and a decryption routine. These three elements all exist within the single executable application, which accordingly runs as a single process when executed. The three components can easily be used merely by JUMP instructions within the executable application and thus speed issues are minimized.

By contrast, Altberg describes an approach which makes use of two separate processes. These are called the "application" and the "installer module". Altberg contains many statements which make clear that these processes are separate. For example:

(column 4, line 36) - "one or more application programs 36, installer module 37";

(column 7, line 66) - refers to the "application" and that "the installer module is launched";

(column 8, line 2) - "after launching the installer ... the application execution process is terminated";

(column 8, line 8) - "the application execution process may be reinitiated after the required files are installed";

(column 10, lines 16 to 22) - "the application calls an installer module ... the application then terminates. The installer module then controls the installation ... the installer will launch the application so that the execution process can be completed".

Accordingly, in Altberg, if execution of the application process reveals that a file is missing, a second process (the installer module) must be called. This makes the Altberg arrangement vulnerable. It is completely inoperable if the installer module is found to be unavailable (for example by the action of some other application which has aggressively changed the system resources), or if the installer module has become corrupt, such as virus or Trojan operation. Embodiments of the present invention do not suffer this vulnerability, because they are self-contained.

Calling a second process, in Altberg, will also incur time penalties as the second process is loaded. Further time penalties are incurred because the second process must then terminate the first process. This is necessary because shared resources such as dll files are locked while a process which requires them is running. Accordingly, "after launching the installer ... the application execution process is terminated" (column 8, line 2).

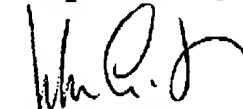
Once the installer module has executed, further time penalties are incurred in reinitiating the application execution process (column 8, line 8) and it may even be necessary to restart the system (column 8, lines 20 to 28, particularly line 26).

By virtue of the self-contained nature of embodiments of the present invention, (i.e. including the program, required sub-routines and decryption routine for conversion of the encrypted sub-routines) only one process is required. This results in improvements in speed (because second processes do not need to be

loaded etc.) and reduced vulnerability (which would arise from second processes not being available). Accordingly, the present invention differs from the disclosure of Altberg in the manner in which the present invention is self-contained, which is nowhere suggested by Altberg. Moreover, this difference provides significant technical advantages of speed and reliability and thus represents an inventive step over the disclosure of Altberg.

In view of the foregoing, it is submitted that the independent claims 1, 10, 11, 12 and 21 are not anticipated by Altberg et al. Applicant submits that the independent claims are patentable. Since applicant has shown that the independent claims are patentable, it follows that the dependent claims also are patentable and applicant is not required to address the rejections of the dependent claims.

Respectfully submitted,



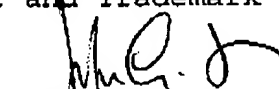
John Smith-Hill
Reg. No. 27,730

SMITH-HILL & BEDELL, P.C.
12670 NW Barnes Road, Suite 104
Portland, Oregon 97229

Tel. (503) 574-3100
Fax (503) 574-3197
Docket: SWIN 2275

Certificate of Facsimile Transmission

I hereby certify that this paper is being facsimile transmitted to the Patent and Trademark Office on the date shown below.



John Smith-Hill

4/4/05

Date

ABSTRACT

When an executable program is to be loaded into RAM 10, the RAM initially contains a loader module 12a, the program 14, an ENGINE 22 and encrypted resources 24, such as encrypted .DLL files. When the program 14 is to be run, the loader module 12a will call the ENGINE 22, which will access the sub-routine resources required by the program 14, identify any of these which are already available in the system, load those already available, and decrypt and load any which are not available. This ensures that the required sub-routines are available to the program 14 on each occasion that the program 14 is executed.